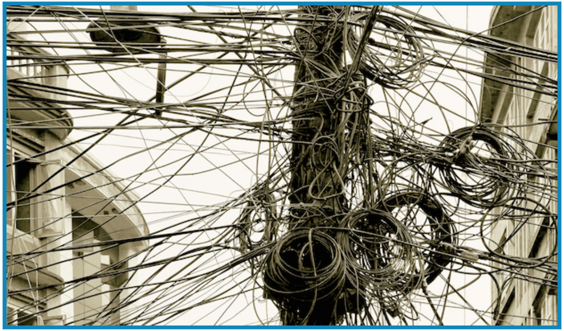
VIS2019

Post-Hoc Analysis with TTK

Jonas Lukasczyk, TU Kaiserslautern

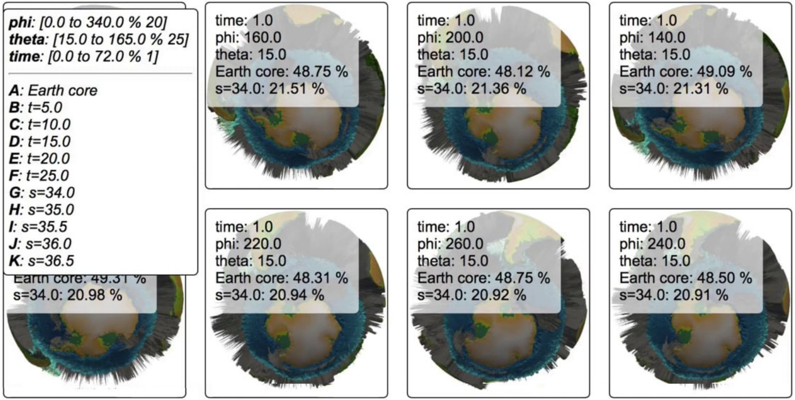
Motivation

- To explore massive datasets, it is necessary to intelligently store, query, analyze, and visualize their data products.
- Such products can be images, meshes, tabular data, tracking graphs, volume data, persistence diagrams, and so forth.



Approach

- Cinema Databases associate data products with parameters.
- Originally conceptualized for image products[1].



phi: [0.0 to 340.0 % 20]
theta: [15.0 to 165.0 % 25]
time: [0.0 to 72.0 % 1]

A: Earth core
B: t=5.0
C: t=10.0
D: t=15.0
E: t=20.0
F: t=25.0
G: s=34.0
H: s=35.0
I: s=35.5
J: s=36.0
K: s=36.5

Earth core: 49.31 %
s=34.0: 20.98 %

time: 1.0
phi: 160.0
theta: 15.0
Earth core: 48.75 %
s=34.0: 21.51 %

time: 1.0
phi: 200.0
theta: 15.0
Earth core: 48.12 %
s=34.0: 21.36 %

time: 1.0
phi: 140.0
theta: 15.0
Earth core: 49.09 %
s=34.0: 21.31 %

time: 1.0
phi: 220.0
theta: 15.0
Earth core: 48.31 %
s=34.0: 20.94 %

time: 1.0
phi: 260.0
theta: 15.0
Earth core: 48.75 %
s=34.0: 20.92 %

time: 1.0
phi: 240.0
theta: 15.0
Earth core: 48.50 %
s=34.0: 20.91 %

[1] Ahrens et al., "An image-based approach to extreme scale in situ visualization and analysis". Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. IEEE Press, 2014.

Cinema Database Concept[2]

Data Product Representation: products can be of any kind, but VTK is strongly recommended.

Data Product Organization: all products are organized by one table that is stored as a CSV file.

Database Interaction: database viewers enable users to specify how database content needs to be interpreted.

[2] D. Rogers et al., ”Cinema Dietrich Specification”. Technical Report, LANL, 2018.

Cinema Database Example

```
.../Meshes.cdb/
    data.csv
    data/
        A_00.vtu
        A_01.vtu
        B_50.vtu
        B_51.vtu
        B_54.vtu
```

(a) File System

```
Sim,   Time,   FILE
A,     00,     data/A_00.vtu
A,     01,     data/A_01.vtu
B,     50,     data/B_50.vtu
B,     51,     data/B_51.vtu
B,     54,     data/B_54.vtu
```

(b) data.csv

New TTK Cinema Filters

General Filters:

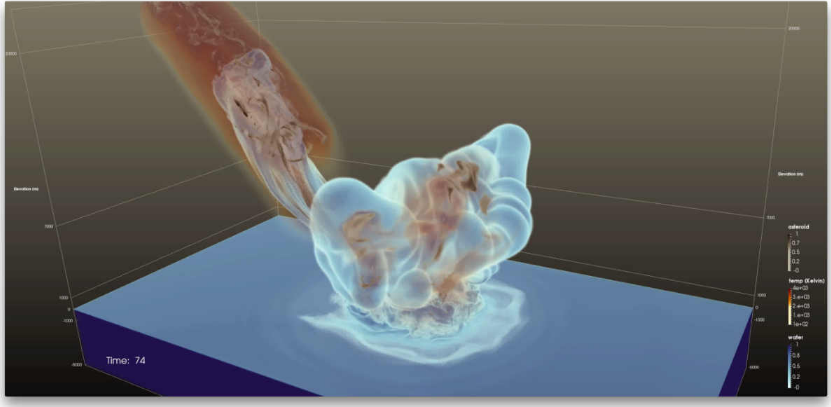
- **CinemaReader** - read database manifest
- **CinemaQuery** - find specific products
- **CinemaProductReader** - read referenced products
- **CinemaWriter** - store products in database

Filters that focus on Image Products:

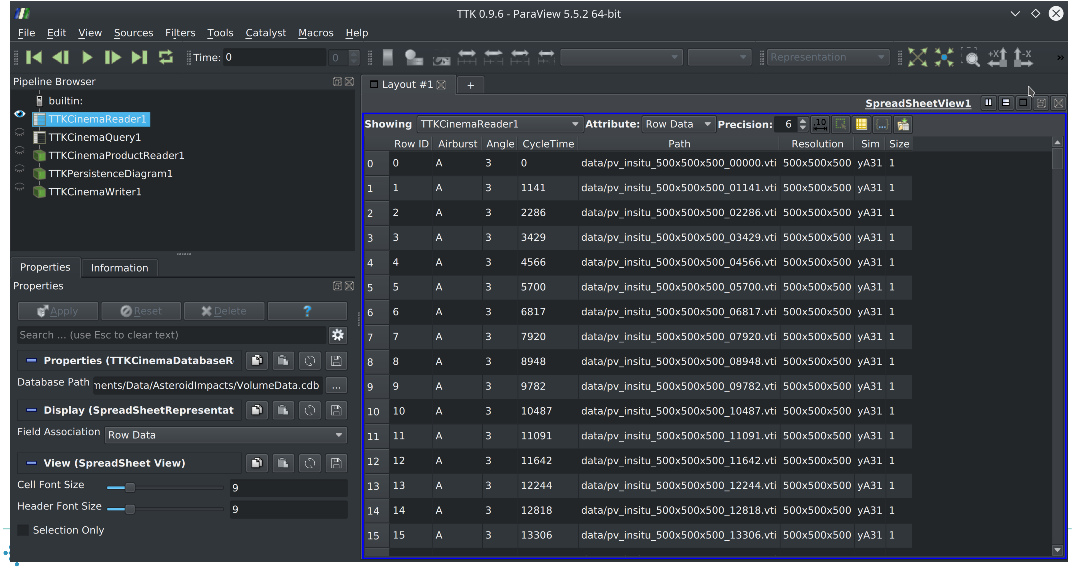
- **CinemaImaging** - create value and depth images of an object
- **CinemaShading** - create color images based on value and depth images

Asteroid Ocean Impacts

- `https://sciviscontest2018.org/`
- 27 different simulation scenarios
- 11 Scalar Fields
- 500^3 cells over 500 timesteps



Reading a Cinema Database



Properties (TTKCinemaDatabaseR

Database Path nents/Data/AsteroidImpacts/VolumeData.cdb

Display (SpreadSheetRepresentat

Field Association Row Data

View (SpreadSheet View)

Cell Font Size 9

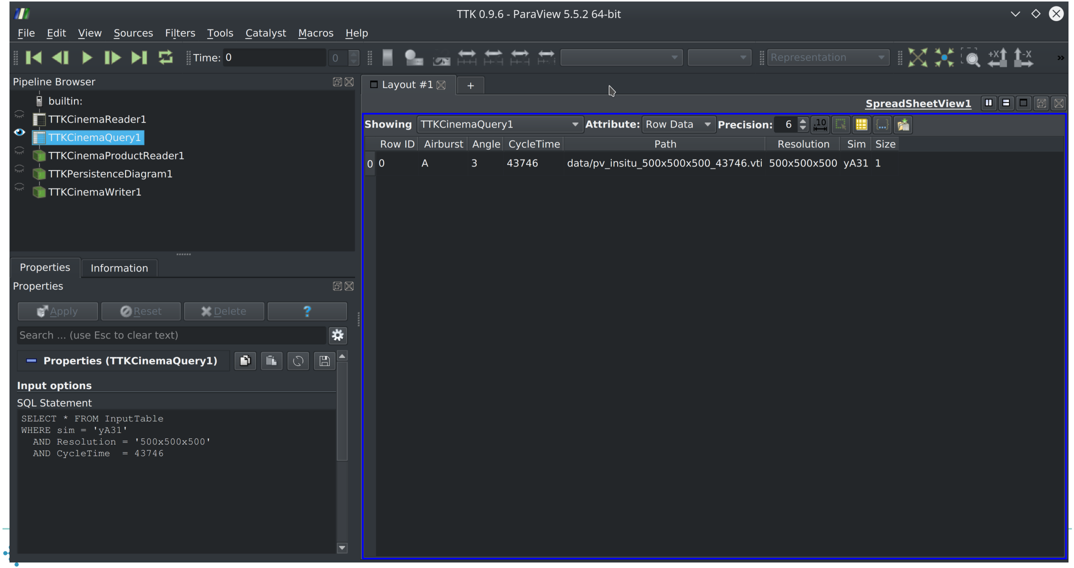
Header Font Size 9

□ Selection Only

Showing TTKCinemaReader1 Attribute: Row Data Precision: 6

	Row ID	Airburst	Angle	CycleTime	Path	Resolution	Sim	Size
0	0	A	3	0	data/pv_insitu_500x500x500_00000.vti	500x500x500	yA31	1
1	1	A	3	1141	data/pv_insitu_500x500x500_01141.vti	500x500x500	yA31	1
2	2	A	3	2286	data/pv_insitu_500x500x500_02286.vti	500x500x500	yA31	1
3	3	A	3	3429	data/pv_insitu_500x500x500_03429.vti	500x500x500	yA31	1
4	4	A	3	4566	data/pv_insitu_500x500x500_04566.vti	500x500x500	yA31	1
5	5	A	3	5700	data/pv_insitu_500x500x500_05700.vti	500x500x500	yA31	1
6	6	A	3	6817	data/pv_insitu_500x500x500_06817.vti	500x500x500	yA31	1
7	7	A	3	7920	data/pv_insitu_500x500x500_07920.vti	500x500x500	yA31	1
8	8	A	3	8948	data/pv_insitu_500x500x500_08948.vti	500x500x500	yA31	1
9	9	A	3	9782	data/pv_insitu_500x500x500_09782.vti	500x500x500	yA31	1
10	10	A	3	10487	data/pv_insitu_500x500x500_10487.vti	500x500x500	yA31	1
11	11	A	3	11091	data/pv_insitu_500x500x500_11091.vti	500x500x500	yA31	1
12	12	A	3	11642	data/pv_insitu_500x500x500_11642.vti	500x500x500	yA31	1
13	13	A	3	12244	data/pv_insitu_500x500x500_12244.vti	500x500x500	yA31	1
14	14	A	3	12818	data/pv_insitu_500x500x500_12818.vti	500x500x500	yA31	1
15	15	A	3	13306	data/pv_insitu_500x500x500_13306.vti	500x500x500	yA31	1

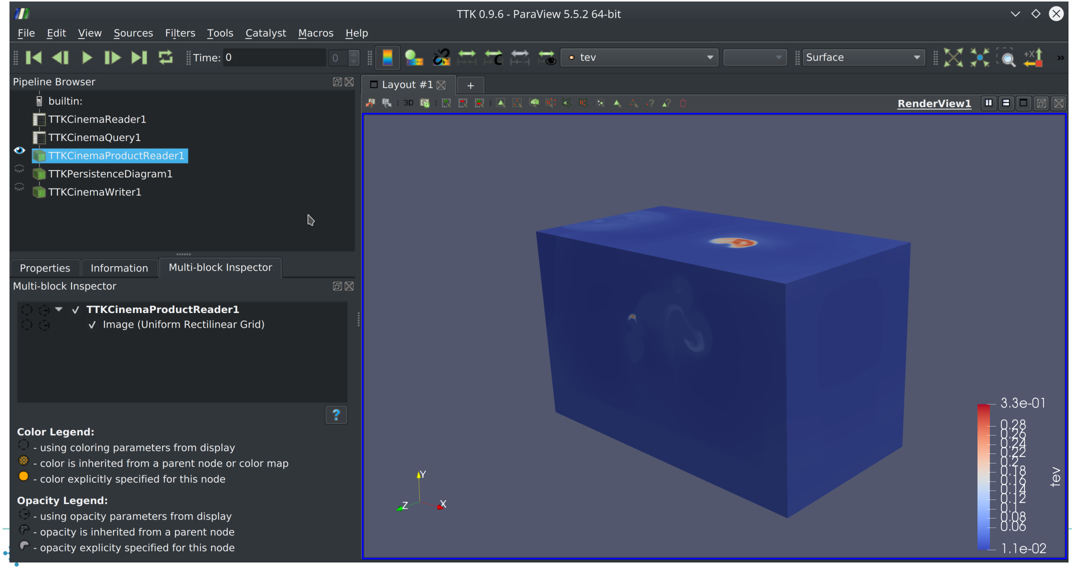
Performing a Query



```
SELECT * FROM InputTable
WHERE sim = 'yA31'
  AND Resolution = '500x500x500'
  AND CycleTime  = 43746
```

	Row ID	Airburst	Angle	CycleTime	Path	Resolution	Sim	Size
0	0	A	3	43746	data/pv_insitu_500x500x500_43746.vti	500x500x500	yA31	1

Reading the corresponding Data Products



Pipeline Browser

- builtin:
- TTKCinemaReader1
- TTKCinemaQuery1
- TTKCinemaProductReader1
- TTKPersistenceDiagram1
- TTKCinemaWriter1

Properties | Information | Multi-block Inspector

Multi-block Inspector

- ✓ **TTKCinemaProductReader1**
  - ✓ Image (Uniform Rectilinear Grid)

?

Color Legend:

○ - using coloring parameters from display

▨ - color is inherited from a parent node or color map

● - color explicitly specified for this node

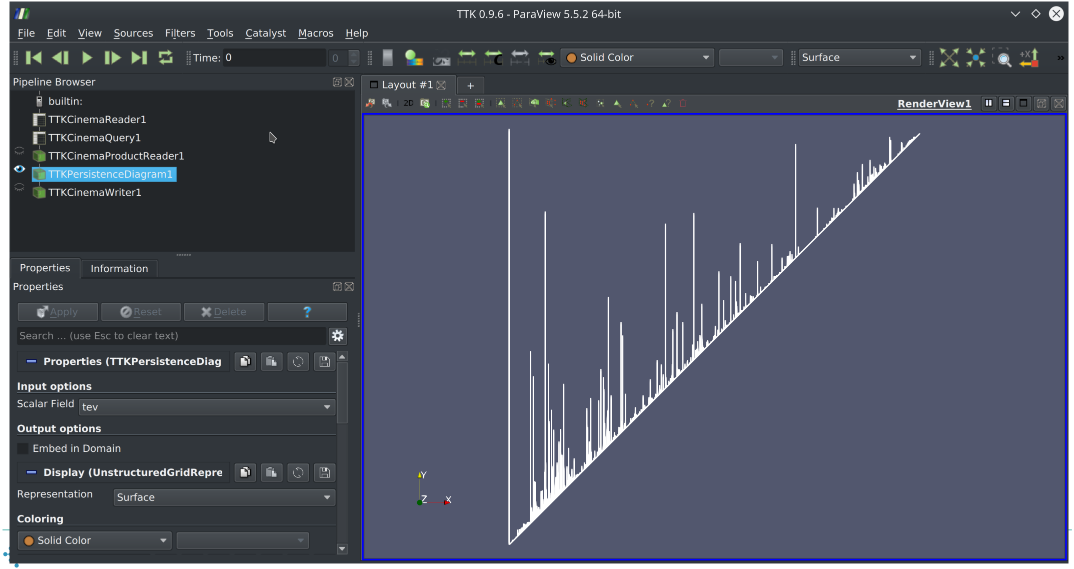
Opacity Legend:

○ - using opacity parameters from display

◔ - opacity is inherited from a parent node

◕ - opacity explicitly specified for this node

Perform any Analysis



TTK 0.9.6 - ParaView 5.5.2 64-bit

Pipeline Browser

- builtin:
- TTKCinemaReader1
- TTKCinemaQuery1
- TTKCinemaProductReader1
- TTKPersistenceDiagram1
- TTKCinemaWriter1

Properties | Information

Properties

Search ... (use Esc to clear text)

Properties (TTKPersistenceDiag

Input options

Scalar Field: tev

Output options

□ Embed in Domain

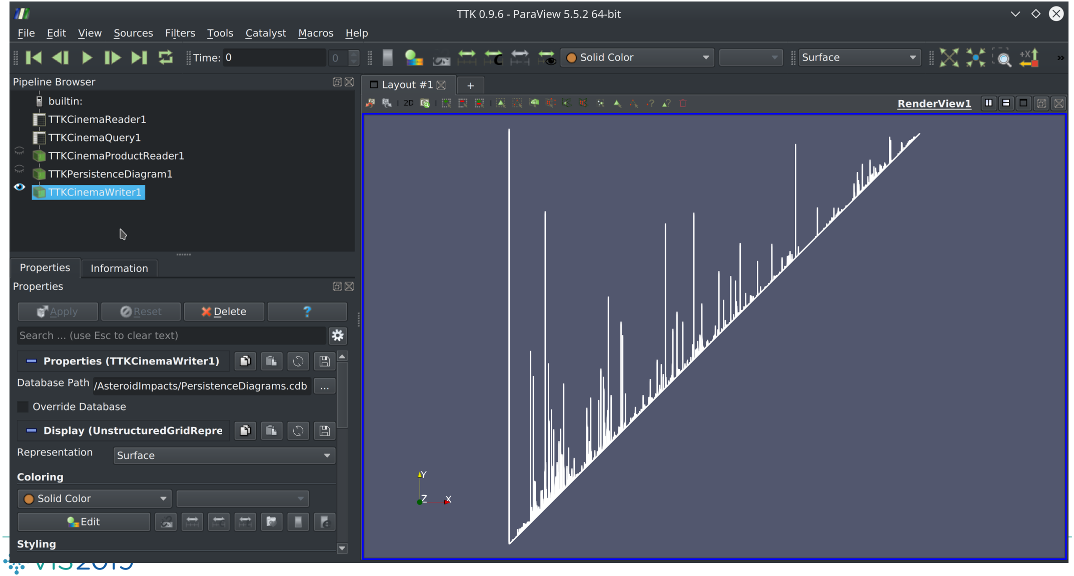
Display (UnstructuredGridRepre

Representation: Surface

Coloring

Solid Color

Store new Data Products



TTK 0.9.6 - ParaView 5.5.2 64-bit

Pipeline Browser

- builtin:
- TTKCinemaReader1
- TTKCinemaQuery1
- TTKCinemaProductReader1
- TTKPersistenceDiagram1
- TTKCinemaWriter1

Properties (TTKCinemaWriter1)

Database Path: /AsteroidImpacts/PersistenceDiagrams.cdb

□ Override Database

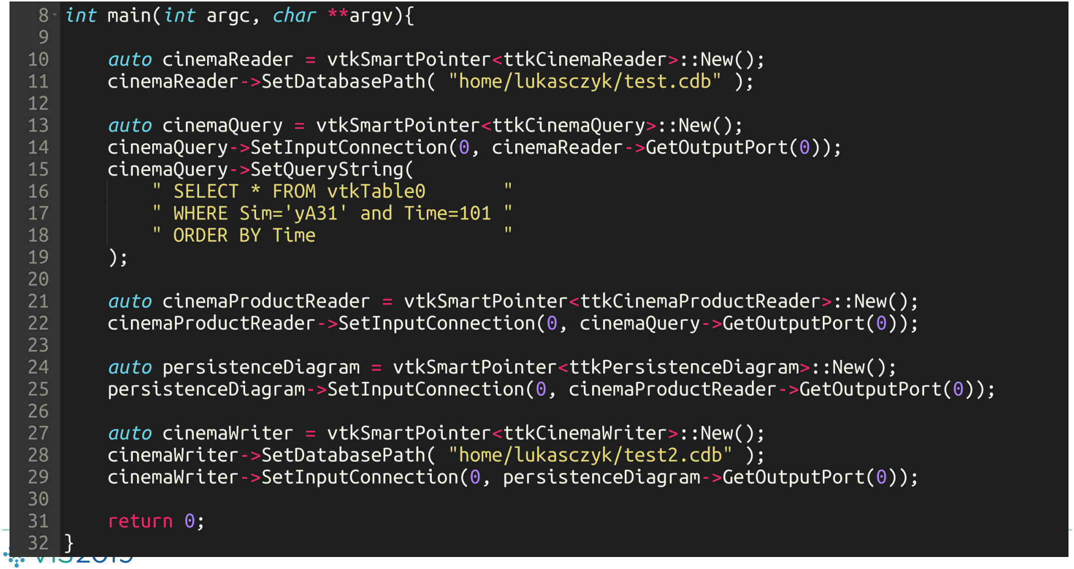
Display (UnstructuredGridRepre

Representation: Surface

Coloring: Solid Color

Styling

C++ Script



```cpp
int main(int argc, char **argv){

    auto cinemaReader = vtkSmartPointer<ttkCinemaReader>::New();
    cinemaReader->SetDatabasePath( "home/lukasczyk/test.cdb" );

    auto cinemaQuery = vtkSmartPointer<ttkCinemaQuery>::New();
    cinemaQuery->SetInputConnection(0, cinemaReader->GetOutputPort(0));
    cinemaQuery->SetQueryString(
        " SELECT * FROM vtkTable0       "
        " WHERE Sim='yA31' and Time=101 "
        " ORDER BY Time                 "
    );

    auto cinemaProductReader = vtkSmartPointer<ttkCinemaProductReader>::New();
    cinemaProductReader->SetInputConnection(0, cinemaQuery->GetOutputPort(0));

    auto persistenceDiagram = vtkSmartPointer<ttkPersistenceDiagram>::New();
    persistenceDiagram->SetInputConnection(0, cinemaProductReader->GetOutputPort(0));

    auto cinemaWriter = vtkSmartPointer<ttkCinemaWriter>::New();
    cinemaWriter->SetDatabasePath( "home/lukasczyk/test2.cdb" );
    cinemaWriter->SetInputConnection(0, persistenceDiagram->GetOutputPort(0));

    return 0;
}
```

Conclusion

- Cinema integration in TTK
- Data analysis and visualization made easy
- Systematic approach to batch processing and post-hoc analysis
- Trivial to get c++/pyhton scripts running on a cluster (no need for ParaView)

Thank You